

BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	SSSSSSSS
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	SSSSSSSS
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	SSSSSSSS
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		00000000		00000000		TTT	SSS	SSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTT	SSS	SSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTT	SSS	SSSSSSSS

```
RRRRRRRR      EEEEEEEEEEE      AAAAAA      DDDDDDDDD      PPPPPPPP      RRRRRRRR      MM      MM      PPPPPPPP      TTTTTTTTTT
RRRRRRRR      EEEEEEEEEEE      AAAAAA      DDDDDDDDD      PPPPPPPP      RRRRRRRR      MM      MM      PPPPPPPP      TTTTTTTTTT
RR      RR      EE      AA      AA      DD      DD      PP      PP      RR      RR      MMMM      MMMM      PP      PP      TT
RR      RR      EE      AA      AA      DD      DD      PP      PP      RR      RR      MMMM      MMMM      PP      PP      TT
RR      RR      EE      AA      AA      DD      DD      PP      PP      RR      RR      MM      MM      PP      PP      TT
RRRRRRRR      EEEEEEEEEEE      AA      AA      DD      DD      PPPPPPPP      RRRRRRRR      MM      MM      PPPPPPPP      TT
RRRRRRRR      EEEEEEEEEEE      AA      AA      DD      DD      PPPPPPPP      RRRRRRRR      MM      MM      PPPPPPPP      TT
RR      RR      EE      AAAAAAAAAA      DD      DD      PP      RRRRRRRR      MM      MM      PP      TT
RR      RR      EE      AAAAAAAAAA      DD      DD      PP      RR      RR      MM      MM      PP      TT
RR      RR      EE      AA      AA      DD      DD      PP      RR      RR      MM      MM      PP      TT
RR      RR      EE      AA      AA      DD      DD      PP      RR      RR      MM      MM      PP      TT
RR      RR      EEEEEEEEEEE      AA      AA      DDDDDDDDD      PP      RR      RR      MM      MM      PP      TT
RR      RR      EEEEEEEEEEE      AA      AA      DDDDDDDDD      PP      RR      RR      MM      MM      PP      TT
```

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```


(2) 52
(3) 89
(3) 162

DECLARATIONS
BOO\$READPROMPT - Prompt and read input string
RIO\$OUTPUT_LINE - Output one line

```

0000 1      .TITLE READPRMPT - READ AND PROMPT ROUTINE
0000 2      .IDENT 'V04-000'
0000 3
0000 4
0000 5      *****
0000 6      *
0000 7      * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      * ALL RIGHTS RESERVED.
0000 10     *
0000 11     * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     * TRANSFERRED.
0000 17     *
0000 18     * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     * CORPORATION.
0000 21     *
0000 22     * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     *
0000 25     *
0000 26     *****
0000 27
0000 28
0000 29     ++
0000 30     FACILITY:
0000 31
0000 32     ABSTRACT:
0000 33     This module contains a routine (BOO$READPROMPT) which writes a
0000 34     prompt line and reads a line of input from the console terminal
0000 35     using QIOs. Either writing the prompt line or reading the input line
0000 36     may be bypassed.
0000 37
0000 38     ENVIRONMENT: User mode
0000 39
0000 40     AUTHOR: STEVE BECKHARDT,      CREATION DATE: 27-Sep-1979
0000 41
0000 42     MODIFIED BY:
0000 43
0000 44     V03-002 KDM0090      Kathleen D. Morse      01-Dec-1983
0000 45     Make psect word aligned.
0000 46
0000 47     V03-001 JLV0134      Jake VanNoy           31-Dec-1981
0000 48     Add routine RIO$OUTPUT_LINE.
0000 49
0000 50     --

```



```

0000 52      .SBTTL  DECLARATIONS
0000 53      :
0000 54      : INCLUDE FILES:
0000 55      :
0000 56      :
0000 57      :
0000 58      : MACROS:
0000 59      :
0000 60      :
0000 61      :
0000 62      : EQUATED SYMBOLS:
0000 63      :
0000 64      :
0000 65      :
0000 66      : OWN STORAGE:
0000 67      :
0000 68      :
00000000 69      .PSECT  BOO$SYSGEN,WRT,WORD
0000 70      :
00000008 0000 71 IOSTBLK:      ; I/O status block
0000 72      .BLKQ  1
0000 73      :
0000 74 CHANNEL:      ; Channel
0000 75      .WORD  0
000A 76      :
30 41 50 4F 5F 00000012'010E0000' 000A 77 DEVNAM_DSC:      ; Device name descriptor
0000 0017 78      .ASCID  /_OPA0/
0000 0019 79      :
00000100 0019 80 RIO$GW_OUTLEN:: .WORD  0
00000021' 001D 81 RIO$AB_OUTBUF::
0000 0021 82      .LONG  256      ; Descriptor
0000 0021 83      .LONG  RIO$AB_BUFFER      ; Buffer pointer
0000 0121 84 RIO$AB_BUFFER::
0000 0121 85      .BLKB  256      ; Buffer
00000000 86      :
0000 87      .PSECT  BOO$READPROMPT,RD,NOWRT,EXE

```

```
0000 89 .SBTTL BOO$READPROMPT - Prompt and read input string
0000 90 :++
0000 91 : Functional Description:
0000 92 : BOO$READPROMPT outputs the specified ASCIIZ prompt string on the
0000 93 : console terminal then checks the count of characters to be read.
0000 94 : If zero it exits, otherwise it reads the console terminal until
0000 95 : either a carriage return is encountered or the character count
0000 96 : is satisfied. The specified buffer is filled with an ASCII
0000 97 : string containing the characters read but not including the
0000 98 : terminating carriage return.
0000 99 :
0000 100 : Calling Sequence:
0000 101 : CALLG  ARGLIST,BOO$READPROMPT
0000 102 :
0000 103 : Input Parameters:
0000 104 : PROMPT(AP) - Address of ASCIIZ prompt string
00000004 0000 105 : PROMPT = 4
0000 106 :
0000 107 : SIZE(AP) - Maximum length of input string
00000008 0000 108 : SIZE = 8
0000 109 : Note: if size is zero, then nothing is read
0000 110 : and only the prompt string is written.
0000 111 :
0000 112 : BUF(AP) - Address of input buffer
0000000C 0000 113 : BUF = 12
0000 114 :
0000 115 : Output Parameters:
0000 116 : R0 - Completion status code
0000 117 :
0000 118 : Buffer located by BUF(AP) will be filled with the string
0000 119 : read as an ASCII string.
0000 120 :
0000 121 :--
0000 122 :
0000 123 BOO$READPROMPT::
0004 0000 124 .WORD ^M<R2>
0002 125
0008'CF B5 0002 126 TSTW W^CHANNEL ; Channel assigned yet?
18 12 0006 127 BNEQ 10$ ; Yes
0008 128 $ASSIGN_S CHAN = W^CHANNEL,- ; No, assign it
0008 129 DEVNAM = DEVNAM_DSC,-
0008 130 ACMODE = #3 ; Allow access from user mode
73 50 E9 001D 131 BLBC R0,90$ ; Error
0020 132
04 BC FFFF 8F 00 3A 0020 133 10$: LOCC #0,#^XFFFF,@PROMPT(AP) ; Locate end of prompt string
51 04 AC C2 0027 134 SUBL PROMPT(AP),R1 ; R1 = size of prompt string
50 08 AC D0 002B 135 MOVL SIZE(AP),R0 ; R0 = size of input buffer
52 0C AC D0 002F 136 BEQL 20$ ; No input buffer
0031 137 MOVL BUF(AP),R2 ; R2 = address of input buffer
0035 138
0035 139 $QIOW_S CHAN = W^CHANNEL,- ; Prompt and read
0035 140 FUNC = #10$ READPROMPT,-
0035 141 IOSB = W^IOSTBLK,-
0035 142 P1 = 1(R2),- ; Address of input buffer
0035 143 P2 = R0,- ; Size of input buffer
0035 144 P5 = PROMPT(AP),- ; Address of prompt buffer
0035 145 P6 = R1 ; Size of prompt buffer
```



```

50      36 50      E9 005A 146      BLBC  R0,90$      : Error
      0000'CF      3C 005D 147      MOVZWL W^IOSTBLK,R0      : Get I/O status block
62      0002'CF      90 0062 148      MOVB  W^IOSTBLK+2,(R2)      : Store size of input line
      2A      11 0067 149      BRB  90$
      0069 150
      0069 151 20$: $QIOW_S CHAN = W^CHANNEL,-      : Write prompt string, no input
      0069 152      FUNC = #IOS$ WRITEVBLK,-
      0069 153      IOSB = W^IOSTBLK,-
      0069 154      P1 = @PROMPT(AP),-      : Address of prompt buffer
      0069 155      P2 = R1      : Size of prompt buffer
50      05 50      E9 008B 156      BLBC  R0,90$      : Error
      0000'CF      3C 008E 157      MOVZWL W^IOSTBLK,R0      : Get I/O status block
      0093 158
      04 0093 159 90$: RET
      0094 160

```

```

0094 162 .SBTTL RIO$OUTPUT_LINE - Output one line
0094 163
0094 164 :+
0094 165 : This routine is in RMSCONIO for SYSGEN, is used here to map STASYSGEN
0094 166 : calls to this routine into calls to BOO$READPROMPT.
0094 167
0094 168 : Inputs:
0094 169 :     RIO$GW_OUTLEN - length of string to output
0094 170 :     RIO$AB_BUFFER - buffer to output
0094 171 :-
0094 172
0094 173 RIO$OUTPUT_LINE::
0094 174
51      7E      51      7D 0094 175      MOVQ      R1,-(SP)          ; Save R1,R2
      00000017'EF 3C 0097 176      MOVZWL     RIO$GW_OUTLEN,R1      ; Set length
52      00000021'EF 9E 009E 177      MOVAB      RIO$AB_BUFFER,R2      ; Set address
      51      6241 9E 00A5 178      MOVAB      (R2)[RT],R1          ; Set address of end of string
61      00000A0D 8F D0 00A9 179      MOVL       #^X00000A0D,(R1)      ; Set CR, LF, zero byte at end
      00B0 180
      7E      7C 00B0 181      CLRQ        -(SP)          ; Null read buffer
      52      DD 00B2 182      PUSHL       R2              ; Address of string
FFFFF45 EF 03 FB 00B4 183      CALLS      #3,L^BOO$READPROMPT      ; Output string
      51      8E 7D 00BB 184
      05 00BE 185      MOVQ        (SP)+,R1          ; Restore R1,R2
      00BF 186      RSB              ; Return
      00BF 187
      00BF 188      .END

```


READPRMPT
Symbol table

- READ AND PROMPT ROUTINE

H 10

15-SEP-1984 23:59:11 VAX/VMS Macro V04-00
4-SEP-1984 23:05:24 [BOOTS.SRC]READPRMPT.MAR;1

Page 6
(3)

```

$$T1 = 00000001
BOOS$READPROMPT = 00000000 RG 02
BUF = 0000000C
CHANNEL = 00000008 R 01
DEVNAM_DSC = 0000000A R 01
IOS_READPROMPT ***** X 02
IOS_WRITEVBLK ***** X 02
IOSTBLK = 00000000 R 01
PROMPT = 00000004
RIOSAB_BUFFER = 00000021 RG 01
RIOSAB_OUTBUF = 00000019 RG 01
RIOSGW_OUTLEN = 00000017 RG 01
RIOSOUTPUT_LINE = 00000094 RG 02
SIZE = 00000008
SYSS$ASSIGN ***** GX 02
SYSS$QIOW ***** GX 02

```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
BOOS\$SYSGEN	00000121 (289.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC WORD
BOOS\$READPROMPT	000000BF (191.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.09	00:00:00.79
Command processing	128	00:00:00.65	00:00:02.88
Pass 1	129	00:00:01.08	00:00:03.45
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	48	00:00:00.45	00:00:01.08
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.01	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	342	00:00:02.31	00:00:08.25

The working set limit was 900 pages.
 4406 bytes (9 pages) of virtual memory were used to buffer the intermediate code.
 There were 10 pages of symbol table space allocated to hold 16 non-local and 3 local symbols.
 188 source lines were read in Pass 1, producing 13 object records in Pass 2.
 6 pages of virtual memory were used to define 6 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	6

70 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:READPRMPT/OBJ=OBJ\$:READPRMPT MSRC\$:READPRMPT/UPDATE=(ENH\$:READPRMPT)+EXECML\$/LIB+LIB\$:BOOTS.MLB/LIB

0039 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY